Dockct No.: 42390.P9429

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Jin Yang Application No. 09/608,637 Filed: June 30, 2000 For: METHODS FOR FORMAL VERIFICATION ON A SYMBOLIC LATTICE DOMAIN	Examiner: E. Garcia Otero Art Unit: 2123 **CERTIFICATE OF TRANSMISSION** I hereby certify that this correspondence is being facsimile transmitted to the United States Patent and Trademark Office, Fax No. (531) 273-8300___ on _7-18-05_ _____ Date Lawrence M. Mennemeier
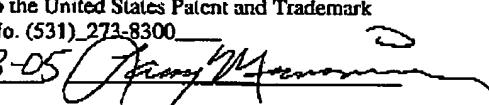
APPELLANT'S BRIEF UNDER 37 CFR § 41.37
IN SUPPORT OF APPELLANT'S APPEAL TO THE BOARD OF PATENT
APPEALS AND INTERFERENCES

Mail Stop Appeal Brief-Patents
Commissioner of Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellant hereby submits this Brief in support of an appeal from a final decision of

the Examiner, in the above-referenced case. Appellant respectfully requests consideration

of this appeal by the board of Patent Appeals and Interference for allowance of the above-

referenced patent application.

-1-

TABLE OF CONTENTS

I.   Real Party in Interest

The real party in interest in the present appeal is Intel Corporation of Santa Clara, California, the assignee of the present application.

II.   Related Appeals and Interferences

There are no related appeals or interferences to appellant's knowledge that would have a bearing on any decision of the Board of Patent Appeals and Interferences.

III.   Status of the Claims (independent claims shown in bold)

Claims 1-3, 6-7, 9-13, 19-20, 21, 22-27 and 29-30 are canceled.

Claims 4-5 stand rejected under 35 USC § 112 as allegedly being indefinite.

Claims 4-5, **8**, 14-15, **16**-18 and **28** stand rejected under 35 USC § 102(b) as allegedly being anticipated by the Ph.D. dissertation of Alok Jain at Carnegie Mellon University, July 1997.

Final rejection of claims 4-5, **8**, 14-15, **16**-18 and **28** is being appealed.

42390.P9429   -3-

IV.   Status of Amendments

An official amendment and response to a first Office Action mailed 3/9/2004 was submitted by appellant on 9/9/2004 and was entered. A Final Office Action was mailed on 12/17/2004. Appellant responded by submitting an amendment and official response after final on 4/18/2005. It is not known whether the amendment submitted on 4/18/2005 was entered. A Notice of Appeal was transmitted on 5/17/2005, and an appeal ensued. Another amendment is being submitted, under 37 CFR § 41.33 and concurrent with the present appeal brief, which encompasses and supersedes the amendment of 4/18/2005.

Accordingly, the claims stand as of the concurrently submitted amendment of 7/18/2005, and are reproduced in clean form in the Claims Appendix.

V.   Summary of Claimed Subject Matter

Appellant's disclosure describes methods for formal verification of circuits and other finite-state systems. Formal definitions and semantics are disclosed for a model of a finite-state system, an assertion graph to express properties for verification, and satisfiability criteria for specification and automated verification of forward implication properties and backward justification properties, the latter of which were formerly not supported through prior verification techniques. A method is also disclosed to compute a simulation relation sequence ending with a simulation relation fixpoint, which can be compared to a consequence labeling for each edge of an assertion graph to verify implication properties and justification properties according to the formal semantics.

A method for representing and verifying assertion graphs symbolically is disclosed that provides an effective alternative for verifying families of properties. A symbolic

42390.P9429   -4-

indexing function provides a way of identifying assignments to Boolean variables with particular scalar cases. Formally defining a class of lattice domains based on symbolic indexing functions, provides an efficient symbolic manipulation technique using binary decision diagrams (BDDs).

Claim 8 sets forth a computer software product including one or more recordable media having executable instructions stored thereon which, when executed by a processing device, causes the processing device to initialize a symbolic simulation relation[1] for an assertion graph[2] on a first symbolic lattice domain[3]; and compute the symbolic simulation

---

[1] "For one possible embodiment, an assertion graph, G, can be defined on a finite noncmpty set of vertices, V, to include an initial vertex, vI; a set of edges, E, having one or more copies of outgoing edges originating from each vertex in V; a label mapping, Ant, which labels an edge, e, with an antecedent Ant(e); and a label mapping, Cons, which labels an edge, e, with a consequence, Cons(e). When an outgoing edge, e, originates from a vertex, v, and terminates at vertex, v', the original vertex, v, is called the head of e (written v = Head(e))" (p. 9, lines 11-18), "define a simulation relation sequence, Sim_i: E→P(S), mapping edges between vertices in G into state subsets in [a model] M as follows:

$Sim_1(e) = Ant(e)$ if Head(e)=vI, otherwise
$Sim_1(e) = \{ \};...$" (p.16, line 24 through p. 17, line 2).

"Box 311 represents initially assigning an empty set to the simulation relation for all edges e in the assertion graph that do not begin at initial vertex vI, and initially assigning Ant(e) to the simulation relation for all edges e that do begin at initial vertex vI." (p. 17, lines 14-17, Fig. 3a, 311) "In block 611, the antecedent sets are strengthened for each edge in the assertion graph." (p. 21, lines 8-9, Fig. 6a, 611) "In block 621, the strengthened antecedent set fixpoint for each edge e (denoted $Ant^*(e)$) in assertion graph G is computed." (p. 21, lines 15-16, Fig. 6b, 621)

[2] "For one embodiment, Figure 1b depicts an assertion graph, 102. The two types of labels used in the assertion graph have the following purposes: an antecedent represents a set of possible pre-existing states and stimuli to a circuit or finite state system to affect its behavior; a consequence represents a set of possible resulting states or behaviors to be checked through simulation of the circuit or finite state system. Antecedent and consequence labels are written as ai/ci for the edges of assertion graph 102." (p. 9, lines 19-25, Fig. 1b) "The abstracted assertion graph G_A is an assertion graph on a lattice domain $(P_A, \subseteq_A)$ having the same vertices and edges as G and for the abstracted antecedent labeling Ant_A and the abstracted consequence labeling $Cons_A$, $Ant_A(e)=A(Ant(e))$ and $Cons_A(e)=A(Cons(e))$ for all edges e in the assertion graphs G_A and G." (p. 24, lines 14-18)

[3] "It will be appreciated that the Union operation and the Intersect operation may also be interpreted as the Join operation and the Meet operation respectively." (p. 17, lines 10-13) "One lattice domain of interest is the set of all subsets of [a finite set of states] S, P(S) along with a subset containment relation, $\subseteq$. The subset containment relation defines a partial order between elements of P(S), with the empty set as a lower bound and S as an upper bound. The set P(S) together with the subset containment relation, $\subseteq$, are called a partially ordered system." (p. 22, lines 7-11) "For one embodiment an abstraction of the lattice domain $(P(S), \subseteq)$ onto a lattice domain $(P, \subseteq_A)$ can be defined by an abstraction function A mapping P(S) onto P... Figure 7 illustrates one embodiment of an abstraction function A." (see p. 22, line 23 through p. 23, line 11; Fig. 7)

42390.P9429   -5-

relation[4] for the assertion graph on the first symbolic lattice domain to verify the assertion graph[5] according to a normal satisfiability criteria[6].

Claim 4 sets forth a computer software product including one or more recordable media having executable instructions stored thereon which, when executed by a processing

---

[4] "$Sim_n(e) = Union (Sim_{n-1}(e), (Union_{for\ all\ e'\ such\ that\ Tail(e')=Head(e)} (Intersect (Ant(e), Post(Sim_{n-1}(c'))))))$,
    for all $n>1$.
In the simulation relation defined above, the nth simulation relation in the sequence is the result of inspecting every state sequence along every I-path of lengths up to n. For any $n>1$, a state s is in the nth simulation relation of an edge e if it is either in the n-1th simulation relation of e, or one of the states in its pre-image set is in the n-1th simulation relation of an incoming edge e', and state s is in the antecedent set of e." (p. 17, lines 3-10) "For one embodiment, Figure 3a illustrates a method for computing the simulation relation for a model and an assertion graph." (see p. 17, line 13 through p. 18, line 12; Fig. 3a, 312-317; Fig. 4) "In block 612, a fixpoint simulation relation is computed using the antecedent strengthened assertion graph." (p. 21, lines 9-11, Fig. 6a, 612) "In block 622, a fixpoint simulation relation set for each edge e (denoted Sim*(e)) is computed using the strengthened antecedents computed for each edge in block 621." (p. 21, lines 16-19, Fig. 6b, 622)

[5] "The assertion graph can be seen as a monitor of the circuit, which can change over time. The circuit is simulated and results of the simulation are verified against consequences in the assertion graph. The antecedent sequence on a path selects which traces to verify against the consequences." (p. 16, lines 18-21) "Comparing the final simulation relation for each edge, with the consequence set for that edge, indicates whether the model 101 strongly satisfies the assertion graph 201." (see p. 18, lines 13-21, Fig. 4) "Finally in block 613, the simulation relation sets are compared to the consequence sets to see if, for each edge, the simulation relation set is a subset of the consequence set, which is the necessary condition for satisfiability." (p. 21, lines 11-13; Fig. 6a, 613) "In block 623, the comparison is performed." (p. 21, line 19, Fig. 6b, 623) "For one embodiment, Figure 8b illustrates a method for implicit normal satisfiability using an abstracted simulation relation." (see p. 26, line 7 through p. 27, line 8; Figs. 8b and 8c)

[6] "Strong satisfiability, however, is inadequate for expressing justification properties, which are causes of effects, rather than effects of causes." (p. 14, lines 4-6) "For one embodiment, a normal semantics for assertion graphs that provides for justification properties may be formally defined." (p. 15, lines 4-5) "To say that a state, s, satisfies an edge, e (denoted by s ⊨ e), means that for every trace, t, starting from s and every path, p, starting from e, trace, t, satisfies path, p, under the consequence edge labeling, Cons, whenever trace, t, satisfies path, p, under the antecedent edge labeling, Ant. To say that the model M satisfies assertion graph G (denoted by M ⊨ G), means that for any edge e beginning at initial vertex vI in G, all states, s, in M satisfy edge e." (p. 15, lines 10-16) "In order to indicate normal satisfiability, a method is needed to propagate future antecedents backwards. For one embodiment, a method can be defined to strengthen the antecedent set of an edge e by intersecting it with the pre-image sets of antecedents on future edges." (p. 18, lines 22-25) "For one embodiment, Figure 3b illustrates a method for computing the strengthened antecedents for an assertion graph." (see p. 19, lines 13 through p. 20, line 12; Figs. 3b and 5a) "Figure 5b shows the final simulation relation resulting from iterations of the method of Figure 3a performed on the antecedent strengthened assertion graph 502 and using model 101. Comparing the final simulation relation labels for each edge, with the consequence set for that edge (as shown in assertion graph 202) indicates whether the model 101 strongly satisfies the strengthened assertion graph 502... but more importantly model 101 satisfies assertion graph 202 according to normal satisfiability as previously defined." (p. 20, lines 13-26; Fig. 5b) "For one embodiment, Figure 6a shows a method for computing the normal satisfiability of an assertion graph by a model." (p. 21, lines 7-8; Fig. 6a) "For one embodiment, Figure 6b illustrates, in finer detail, a method of computing normal satisfiability." (p. 21, lines 14-15, Fig. 6b) "Similarly, if methods herein previously disclosed determine that an abstracted model M_A satisfies a true abstraction G_A, then the original model M satisfies the original assertion graph G, according to the normal satisfiability criteria." (p. 25, lines 1-4)

42390.P9429   -6-

device, causes the processing device to initialize a symbolic simulation relation [1,7] for an assertion graph[8] on a first symbolic lattice domain[3,9], wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property [6,10] to verify by computing the symbolic simulation relation[4,5,11].

Claim 14 sets forth a method comprising: initializing a symbolic simulation relation[1,7] for an assertion graph[8] on a first symbolic lattice domain[3,9], wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property[6,10] to verify through computing the symbolic simulation relation[4,5,11].

Claim 16 sets forth a method comprising specifying a justification property with an assertion graph[1,6,7,8,10].

Claim 28 sets forth a verification system comprising: means for initializing a symbolic simulation relation[1,7] for an assertion graph[8,12] on a first symbolic lattice

---

[7] "For an assertion graph G and a model M=(Pre, Post), define an antecedent strengthening sequence, Ant_n: E→P(S), mapping edges between vertices in G into state subsets in M as follows:

$Ant_1(e) = Ant(e)$, and

$Ant_n(e) =$ Intersect $(Ant_{n-1}(e), (Union_{for\ all\ e'\ such\ that\ Head(e')=Tail(e)}\ Pre(Ant_{n-1}(e'))\ ))$, for all n>1.
In the antecedent strengthening sequence defined above, a state s is in the nth antecedent set of an edge e if it is a state in the n-1th antecedent set of e, and one of the states in a pre-image set of the n-1th antecedent set of an outgoing edge e'." (p. 19, lines 5-14) "For one embodiment, Figure 3b illustrates a method for computing the strengthened antecedents for an assertion graph." (see p. 19, line 17 through p. 20, line 4; Fig. 3b)

[8] "As an example of a justification property, one might wish to assert the following: if the system enters state s1, and does not start in state s1, then at the time prior to entering state s1, the system must have been in state s0. For one embodiment, Figure 2b depicts an assertion graph 202, which attempts to capture the justification property asserted in the above example." (p. 14, lines 6-11; Fig. 2b) "The abstracted assertion graph G_A is an assertion graph on a lattice domain $(P_A, \subseteq_A)$ having the same vertices and edges as G and for the abstracted antecedent labeling Ant_A and the abstracted consequence labeling $Cons_A$, $Ant_A(e)=A(Ant(e))$ and $Cons_A(e)=A(Cons(e))$ for all edges e in the assertion graphs G_A and G." (p. 24, lines 14-18)

[9] "Again, it will be appreciated that the Union operation and the Intersect operation may also be interpreted as the Join operation and the Meet operation respectively." (p. 19, lines 14-16)

[10] "For example, Figure 5a shows iterations of antecedent strengthening of graph 202 on model 101." (see p. 20, lines 4-12; Fig. 2, 202 and Fig. 5b, 502)

[11] "Figure 5b shows the final simulation relation resulting from iterations of the method of Figure 3a performed on the antecedent strengthened assertion graph 502 and using model 101." (see p. 20, lines 13-26; Fig. 5b) "In block 622, a fixpoint simulation relation set for each edge e (denoted Sim*(e)) is computed using the strengthened antecedents computed for each edge in block 621." (see p. 21, line 7 through p. 22, line 2; Figs. 6a and 6b)

[12] "Formally defining a class of lattice domains based on symbolic indexing functions, provides an efficient symbolic manipulation technique using BDDs. Therefore previously disclosed methods for antecedent strengthening, abstraction, computing simulation relations, verifying satisfiability and implicit satisfiability may be extended to assertion graphs that are symbolically represented." (p. 27, line 26 through p. 28,

42390.P9429   -7-

domain[3,9,13], wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property[6,10] to verify through computing the symbolic simulation relation[4,5,11,14]; means for computing the symbolic simulation relation[4,11,14] for the assertion graph[8,12] on the first symbolic lattice domain[3,9,13]; and means for checking the symbolic simulation relation[5] to verify a plurality of properties expressed by a plurality of corresponding assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

Claim 5 sets forth the computer software product recited in Claim 4 which, when executed by a processing device, further causes the processing device to compute the symbolic simulation relation[4,11,14] for the assertion graph[8,12] on the first symbolic lattice domain[3,9,13]; and check the symbolic simulation relation[5] to verify a plurality of properties expressed by a plurality of assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

Claim 15 sets forth the method recited in Claim 14 further comprising computing the symbolic simulation relation[4,11,14] for the assertion graph[8,12] on the first symbolic lattice domain[3,9,13]; and checking the symbolic simulation relation[5] to verify a plurality of properties expressed by a plurality of corresponding assertion graph instances, having at

---

line 5) "For one embodiment, an assertion graph G_S on a symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_S$) can be set forth as a mapping $G_S(\underline{b})$ of m-ary boolean values $\underline{b}$ in B^m to scalar instances of assertion graph G_S on the original lattice domain (P, $\subseteq$) such that for the symbolic antecedent labeling Ant_S and the symbolic consequence labeling $Cons_S$,

$Ant_S(\underline{b})(e) = Ant_S(e)(\underline{b})$, and

$Cons_S(\underline{b})(e) = Cons_S(e)(\underline{b})$,

for all edges e in the assertion graph G_S. Figure 11a shows two assertion graphs, 1101 and 1102, on a lattice domain (P, $\subseteq$) and an assertion graph 1103 on the unary symbolic lattice domain 901 that symbolically encodes assertion graphs 1101 and 1102." (see p. 29, line 11 through p. 30, line 19; Fig. 11a)

[13] "For one embodiment, an m-ary symbolic extension of a lattice domain (P, $\subseteq$) can be set forth as a set of symbolic indexing functions $\{B^m \to P\}$ where B^m is the m-ary Boolean product." (p. 28, lines 6-8) "As an example of a symbolic lattice domain, Figure 9 depicts part of a unary symbolic lattice domain." (see p. 28, line 24 through p. 29, line 4; Fig. 9)

[14] "For one embodiment, Figure 12a illustrates a method for computing the simulation relation for a model and an assertion graph on the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_S$)." (see p. 30, line 20 through p. 33, line 12; Figs. 11b and 12a)

42390.P9429   -8-

least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

Claim 18 sets forth The method recited in Claim 17 further comprising computing a symbolic simulation relation[4,11,14] for the assertion graph[8,12] on the first symbolic lattice domain[3,9,13]; and checking the symbolic simulation relation with a symbolic consequence labeling for the assertion graph[5] on the first symbolic lattice domain according to a normal satisfiability criteria[6].

42390.P9429   -9-

VI.   Grounds of Rejection to be Reviewed on Appeal

A. Claims 4-5 stand rejected under 35 USC § 112 as allegedly being indefinite.

B. Claims 4-5, 8, 14-15, 16-18 and 28-28 stand rejected under 35 USC § 102(b)

as allegedly being anticipated by the Ph.D. dissertation of Alok Jain at Carnegie

Mellon University, July 1997.

VII.   Argument

A. 35 U.S.C. § 112 REJECTIONS

Claims 4-5 stands rejected under 35 USC § 112, second paragraph, as allegedly

being indefinite, the Final Office Action (23) stating that it is not clear how the term

"initialize" in independent claim 1 is distinct from or broader than the term "compute."

1. Claims 4 Is Not Indefinite.

The issue of definiteness is whether, in light of the teachings of the prior art and

of the particular invention, the claims set out and circumscribe a particular area with a

reasonable degree of precision and particularity. *In re Moore*, 439 F.2d 1232, 1235, 169

USPQ 236, 238 (CCPA 1971).

Claim 4, for example, sets forth:

> 4.   (Previously Presented) A computer software product including one or more recordable
> media having executable instructions stored thereon which, when executed by a
> processing device, causes the processing device to:
>   initialize a symbolic simulation relation for an assertion graph on a first
> symbolic lattice domain, wherein the assertion graph on the first symbolic lattice
> domain is configurable to express a justification property to verify by computing
> the symbolic simulation relation.

42390.P9429   -10-

The amount of detail required to be included in claims depends on the particular invention and the prior art, and is not to be viewed in the abstract but in conjunction with whether the specification is in compliance with the first paragraph of section 112. *Chemcast Corp. v. Arco Industries Corp.*, 854 F.2d 1328 (Fed. Cir. 1988).

Appellant respectfully submits that the specification has set forth a full and clear description of the claimed subject matter in sufficient detail to support a conclusion by one skilled in the art that Appellent had possession of the claimed invention and further, to enable one skilled in the art to make and use the claimed invention. For example, with regard to initializing a symbolic simulation relation, the specification discloses (p. 9, lines 11-18) that:

> For one possible embodiment, an assertion graph, G, can be defined on a finite nonempty set of vertices, V, to include an initial vertex, vI; a set of edges, E, having one or more copies of outgoing edges originating from each vertex in V; a label mapping, Ant, which labels an edge, e, with an antecedent Ant(e); and a label mapping, Cons, which labels an edge, e, with a consequence, Cons(e). When an outgoing edge, e, originates from a vertex, v, and terminates at vertex, v', the original vertex, v, is called the head of e (written v = Head(e)).

and further discloses (p. 16, line 22 through p. 17, line 2) that:

> For one embodiment, a simulation relation sequence can be defined for model checking according to the strong satisfiability criteria defined above. For an assertion graph G and a model M=(Pre, Post), define a simulation relation sequence, Sim_n: $E{\rightarrow}P(S)$, mapping edges between vertices in G into state subsets in M as follows:
> $Sim_1(e) = Ant(e)$ if $Head(e){=}vI$, otherwise
> $Sim_1(e) = \{ \}; ...$

and further discloses (p. 17, lines 14-17; Fig. 3a, 311) that:

> Box 311 represents initially assigning an empty set to the simulation relation for all edges e in the assertion graph that do not begin at initial vertex vI, and initially assigning Ant(e) to the simulation relation for all edges e that do begin at initial vertex vI.

It will be appreciated that there is a direct correspondence between the formal definition of $Sim_1(e)$ and the initialization performed by Box 311 of Figure 3a. Appellant respectfully submits that at least in light of the above disclosure set forth by the

42390.P9429   -11-

specification, the claims set out and circumscribe initializing a symbolic simulation

relation for symbolic model checking with a reasonable degree of precision and

particularity. The specification further discloses (p. 19, lines 5-14) that:

> For an assertion graph G and a model M=(Pre, Post), define an antecedent strengthening sequence, Ant_n: E→P(S), mapping edges between vertices in G into state subsets in M as follows:
> $Ant_1(e) = Ant(e)$, and
> $Ant_n(e) = $ Intersect $(Ant_{n-1}(e), (Union_{for\ all\ e'\ such\ that\ Head(e')=Tail(e)}\ Pre(Ant_{n-1}(e'))))$, for all $n>1$.
> In the antecedent strengthening sequence defined above, a state s is in the nth antecedent set of an edge e if it is a state in the n-1th antecedent set of e, and one of the states in a pre-image set of the n-1th antecedent set of an outgoing edge e'.

and further discloses (see p. 19, line 17 through p. 20, line 4; Fig. 3b) that:

> For one embodiment, Figure 3b illustrates a method for computing the strengthened antecedents for an assertion graph.

and further discloses (p. 21, lines 15-16; Fig 6b, 621) that:

> In block 621, the strengthened antecedent set fixpoint for each edge e (denoted $Ant^*(e)$) in assertion graph G is computed.

Appellant respectfully submits that at least in light of the above disclosure set

forth by the specification, the claims set out and circumscribe initializing a symbolic

simulation relation for an assertion graph configurable to express a justification property

with a reasonable degree of precision and particularity.

With regard to initializing a symbolic simulation relation for an m-ary symbolic

extension of a lattice domain, the specification further discloses (p. 30, line 20 through p.

31 line 1) that:

> Given a model M_S on the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_S$), and an assertion graph G_S on the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_S$) having edges $(\underline{v}, \underline{v}')$ and $(\underline{v}', \underline{v})$ where $\underline{v}'$ denotes the successors of $\underline{v}$, and $\underline{v}'$ denotes the predecessors of $\underline{v}$, a method to symbolically compute the simulation relation sequence of G_S can be formally defined. For one embodiment, a symbolic simulation relation sequence $Sim_S(\underline{v}, \underline{v}')$ can be defined for model checking according to the strong satisfiability criteria as follows:
> $Sim_{S1}(\underline{v}, \underline{v}') = (initE(\underline{v}, \underline{v}')\ AND\ U)\ Meet_S\ Ant_S(\underline{v}, \underline{v}')$
> where initE is a Boolean predicate for the set of edges outgoing from vL

and further discloses (p. 31, lines 10-15; Fig. 12a, 1211) that:

Box 1211 represents initially assigning

$$Z = (initE(\underline{v}, \underline{v}') \wedge U) \cap_S Ant_S(\underline{v}, \underline{v}')$$

to the simulation relation for all edges $(\underline{v}, \underline{v}')$ in the assertion graph that do not begin at initial vertex vI, and initially assigning

$$Ant_S(\underline{v}, \underline{v}') = (initE(\underline{v}, \underline{v}') \wedge U) \cap_S Ant_S(\underline{v}, \underline{v}')$$

to the simulation relation for all edges $(\underline{v}, \underline{v}')$ that do begin at initial vertex vI.

It will also be appreciated that there is a direct correspondence between the formal definition of $Sim_{Si}(\underline{v}, \underline{v}')$ and the initialization performed by Box 1211 of Figure 12a. Appellant respectfully submits that at least in light of the above disclosure set forth by the specification, the claims set out and circumscribe initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain with a reasonable degree of precision and particularity. The specification further discloses (p. 33, lines 12-17) that:

For one embodiment, an antecedent strengthening sequence $Ant_S(\underline{v}^{\,\cdot}, \underline{v})$ can be defined for model checking according to the normal satisfiability criteria as follows:

$$Ant_{S1}(\underline{v}^{\,\cdot}, \underline{v}) = Ant_S(\underline{v}^{\,\cdot}, \underline{v}), \text{ and}$$

$$Ant_{Sn}(\underline{v}^{\,\cdot}, \underline{v}) = Meet_S (Ant_{Sn-1}(\underline{v}^{\,\cdot}, \underline{v}), (Join_{S \text{ for all } \underline{b} \text{ in } B_m} Pre_S(Sim_{Sn-1}(\underline{v}, \underline{v}'))[\underline{b}/\underline{v}'])), \text{ for all } n>1.$$

and further discloses (see p. 33, line 18 through p. 34, line 9; Fig. 12b) that:

For one embodiment, Figure 12b illustrates a method for computing the strengthened antecedents for an assertion graph on a symbolic lattice domain.

Appellant respectfully submits that at least in light of the above disclosure set forth by the specification, the claims set out and circumscribe, with a reasonable degree of precision and particularity, initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property.

The Final Office Action (10) says that, "the cited portions [of the specification] state 'one possible embodiment... one embodiment... by way of example... by way of example...' and so forth. None of these cited portions provide a clear and definite definition for the term 'initialize'."

Appellant respectfully submits that the specification intentionally discloses

42390.P9429   -13-

numerous examples and embodiments using words, structures, figures, diagrams and formulas to fully set forth the claimed invention to those skilled in the art. Even so, everyone of skill in the art is not necessarily expected to embrace each and every aspect of the invention, but yet, in any particular area of skill relevant to the invention, they should understand the term "initialize" in light of what is set forth in the specification.

The test for definiteness under 35 U.S.C. § 112 is whether those skilled in the art would understand what is claimed when the claim is read in light of the specification. *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1576, 1 USPQ2d, 1081, 1088 (Fed. Cir. 1986).

Therefore, Appellant respectfully submits that in light of the specification, those skilled in the art would understand what is claimed by the limitation, "initialize a symbolic simulation relation for an assertion graph on a first symbolic lattice domain."

42390.P9429   -14-

2. Claims 5 Is Not Indefinite.

Claim 5, for example, sets forth:

5.   (Original) The computer software product recited in Claim 4 which, when executed by a processing device, further causes the processing device to:
    compute the symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and
    check the symbolic simulation relation to verify a plurality of properties expressed by a plurality of assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

With regard to computing the symbolic simulation relation, the specification discloses (p.16, line 22 through p. 17, line 12) that:

For one embodiment, a simulation relation sequence can be defined for model checking according to the strong satisfiability criteria defined above. For an assertion graph G and a model M=(Pre, Post), define a simulation relation sequence, Sim_n: E→P(S), mapping edges between vertices in G into state subsets in M as follows:
    $Sim_1(e)$ = Ant(e) if Head(e)=vJ, otherwise
    $Sim_1(e)$ = { };
    $Sim_n(e)$ =   Union $(Sim_{n-1}(e)$,
         $(Union_{for\ all\ e'\ such\ that\ Tail(e')=Head(e)}$ (Intersect (Ant(e), Post($Sim_{n-1}(e')$))))), for all n>1.
In the simulation relation defined above, the nth simulation relation in the sequence is the result of inspecting every state sequence along every I-path of lengths up to n. For any n>1, a state s is in the nth simulation relation of an edge e if it is either in the n-1th simulation relation of e, or one of the states in its pre-image set is in the n-1th simulation relation of an incoming edge e', and state s is in the antecedent set of e. It will be appreciated that the Union operation and the Intersect operation may also be interpreted as the Join operation and the Meet operation respectively.

and further discloses (p. 17, line17 through p. 18. line 2; Fig. 3a, 312-317) that:

Box 312 represents marking all edges in the assertion graph active. Box 313 represents testing the assertion graph to identify any active edges. If no active edges are identified, then the method is complete. Otherwise, an active edge, e, is selected and marked not active as represented by box 314. Box 315 represents recomputing the simulation relation for edge, e, by adding to the simulation relation for edge e, any states which are in both the antecedent set for edge e and the post-image set for the simulation relation of any incoming edge, e', to e. Box 316 represents testing the simulation relation for edge e to determine if it was changed by the recomputation. If it has changed, all outgoing edges from e are marked as active, as represented by Box 317. In any case, the method flow returns to the test for active edges represented by Box 313.

It will be appreciated that there is a direct correspondence between the formal

definition of $Sim_n(e)$ and the iterative computing performed by Box 315 of Figure 3a.

The specification further discloses (p. 20, lines13-15; Fig. 5b) that:

Figure 5b shows the final simulation relation resulting from iterations of the method of Figure 3a

42390.P9429   -15-

performed on the antecedent strengthened assertion graph 502 and using model 101.

Appellant respectfully submits that at least in light of the above disclosure set forth by the specification, the claims set out and circumscribe computing a symbolic simulation relation for symbolic model checking with a reasonable degree of precision and particularity.

With regard to computing a symbolic simulation relation for an m-ary symbolic extension of a lattice domain, the specification further discloses (p. 30, line 20 through p. 31 line 7) that:

Given a model M_S on the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_s$), and an assertion graph G_S on the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_s$) having edges (v, v') and (v^-, v) where v' denotes the successors of v, and v^- denotes the predecessors of v, a method to symbolically compute the simulation relation sequence of G_S can be formally defined. For one embodiment, a symbolic simulation relation sequence $Sim_S(v, v')$ can be defined for model checking according to the strong satisfiability criteria as follows:

$Sim_{S1}(v, v') = (initE(v, v') \text{ AND } U) \text{ Meet}_S \text{ Ant}_S(v, v')$

where initE is a Boolean predicate for the set of edges outgoing from vl, , and

$Sim_{Sn}(v, v') = Join_S (Sim_{Sn-1}(v, v'), (Join_{S \text{ for all } b \text{ in } Bm} ($

$Meet_S (Ant(v, v'), Post_S(Sim_{Sn-1}(v^-, v))))[b/v^-]))$, for all $n > 1$

where $Join_S$ and $Meet_S$ are the join, $\cup_S$, and meet, $\cap_S$, operators for the symbolic lattice domain ($\{B^m \to P\}$, $\subseteq_s$) and $[b/v^-]$ denotes replacing each occurrence of v^- in the previous expression with b.

and further discloses (p. 31, lines 16-24; Fig. 12a, 1215-1216) that:

Box 1215 represents recomputing the simulation relation for edge (v, v') by adding to the simulation relation for edges (v, v'), any states which are in both the antecedent set for edges (v, v') and the post-image set for the simulation relation of any incoming edges (v^-, v) to (v, v') produced by substituting any b in B^m for v^-. Box 1216 represents testing the simulation relation labeling for edges (v, v') to determine if it was changed by the recomputation. If it has changed, the method flow returns to the recomputation of simulation relation for edges (v, v'), represented by Box 1215. Otherwise a fixpoint has been reached and the method terminates at box 1216.

It will also be appreciated that there is a direct correspondence between the formal definition of $Sim_{Sn}(v, v')$ and the iterated computing performed by Box 1215 of Figure 12a. Appellant respectfully submits that at least in light of the above disclosure set forth by the specification, the claims set out and circumscribe computing the symbolic simulation relation for the assertion graph on the first symbolic lattice domain with a

42390.P9429   -16-

reasonable degree of precision and particularity.

As relied upon above with regard to claim 4, the test for definiteness under 35 U.S.C. § 112 is whether those skilled in the art would understand what is claimed when the claim is read in light of the specification. *Orthokinetics, Inc., supra.*

Appellant respectfully submits that in light of the specification, those skilled in the art would understand what is claimed by the limitation, "compute the symbolic simulation relation for the assertion graph on the first symbolic lattice domain."

B. 35 U.S.C. § 102(b) REJECTIONS

Claims 4-5, 8, 14-15, 16-18 and 28 stand rejected under 35 USC § 102(b) as allegedly being anticipated by, "Formal Hardware Verification by Symbolic Trajectory Evaluation," the Ph.D. dissertation of Alok Jain at Carnegie Mellon University, July 1997 ("Jain").

Appellant respectfully notes that the present application refers to Jain in the Background of the Invention and contrasts the proposed methodology of Jain with embodiments of the present invention in the Detailed Description, pointing out open problems and limitations of Jain and disclosing novel embodiments that provide solutions to those problems and limitations. Accordingly, Appellant submits that claims 4-5, 8, 14-15, 16-18 and 28 are not anticipated by Jain, and offers the following detailed arguments.

1. Claim 8 Is Not Anticipated by Jain.

The MPEP § 2131 states that:

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Appellant respectfully submits that in the cited reference, each and every element as set forth in claim 8 is not found, either expressly or inherently described.

The Final Office Action (38) suggests that the limitations of claim 8 are disclosed by Jain in the Abstract (p. iii) and in the Introduction (p.3).

42390.P9429   -18-

Appellant respectfully disagrees. Claim 8, for example, sets forth:

8.   (Previously Presented) A computer software product including one or more recordable
        media having executable instructions stored thereon which, when executed by a
        processing device, causes the processing device to:
                initialize a symbolic simulation relation for an assertion graph on a first
        symbolic lattice domain; and
                compute the symbolic simulation relation for the assertion graph on the first
        symbolic lattice domain to verify the assertion graph according to a normal
        satisfiability criteria.

The dissertation of Jain relates to a methodology for formal verification using

symbolic trajectory evaluation (§9.1, par. 1) Generally, a trajectory may be accepted,

rejected, or a "don't care." A trajectory is accepted if there is a path such that the

trajectory satisfies the action and reaction formulas along the path. A trajectory is a

"don't care" if there is no path such that the trajectory satisfies the action formulas. The

trajectory is rejected if there is no path that causes the trajectory to be accepted, and there

is a path such that the action formulas are satisfied, but the reaction formulas are not

satisfied (§6.3, Def. 8, par. 2).

Jain refers to this general form of action/reaction assertion as a *prescient*

trajectory assertion, yet Jain limits his verification algorithms instead to what he refers to

as *oblivious trajectory assertions* (§6.3, Def. 9, par. 2, emphasis supplied). According to

Jain's classification of trajectory assertions, the prescient trajectory assertion is the most

expressive and complex, while the oblivious trajectory assertion is the least expressive

and complex (§5.3.2, par. 2).

Upon close inspection it may be appreciated that Jain's relaxation algorithm for

verifying oblivious trajectory assertions (§5.3.3, par. 2, Fig. 5.2) is substantially similar to

the method for computing the simulation relation disclosed with regard to Fig. 3a of the

present application. For example, lines 10 and 12 of Fig. 5.2 describing Jain's relaxation

42390.P9429   -19-

algorithm may be compared with Box 315 of Fig. 3a of the present application, and the recursive invocation of **relax**(G, w_i) following line 12 of Fig. 5.2 may be compared with Box 317 of Fig. 3a.

Jain says that the least fixed point computation corresponds to performing a reachability analysis on the set of node assignments. The relaxation algorithm starts with the source vertex and works its way to the sink vertex (§5.3.3, par. 2).

As the present specification describes it, the satisfiability of oblivious trajectory assertions proposed by Jain may be referred to as "strong satisfiability," where effects (reactions) are checked against corresponding past and present causes (actions) that have been satisfied by a trajectory. With regard to the methodologies of Jain, the present specification discloses, for example (p. 12, line 23 through p. 13, line 3, emphasis supplied) that:

> Strong satisfiability as defined above formally captures a semantics substantially similar to that used in STE and GSTE as proposed in 1997 by Alok Jain. It requires that a consequence hold based solely on past and present antecedents. Strong satisfiability expresses properties that are effects of causes.

The present application further discloses, what it refers to as a "normal satisfiablility," which does not require such strong assumptions with regard to assertion graphs and corresponds more closely to verifying what Jain refers to as the prescient trajectory assertions. For example, the present application discloses (p. 12, lines 18-22) that:

> [I]t shall be demonstrated herein that it is desirable for the semantics to consider all transitions along an infinite path to see if the antecedents are satisfied. If any of the antecedents along an infinite path are violated, then it is not necessary to check the consequences for that path.

The present application further discloses (p. 18, line 23 through p. 19, line 2) that:

> In order to indicate normal satisfiability, a method is needed to propagate future antecedents backwards. For one embodiment, a method can be defined to strengthen the antecedent set of an

42390.P9429   -20-

edge e by intersecting it with the pre-image sets of antecedents on future edges. Since the strengthening method can have rippling effects on the incoming edges to c, the method should be continued until no remaining antecedents can be propagated backwards.

Jain does not discuss or suggest an algorithm (e.g. as disclosed with regard to Fig. 3b of the present application) to propagate future antecedents backwards in order to indicate normal satisfiability. Nor does Jain propose future work or an algorithm extension that would be feasible in dealing with such an expressive satisfiability criteria as the normal satisfiability set forth in claim 8.

Therefore, Appellant respectfully submits that in the cited reference, instructions which cause a processing device to, "compute a symbolic simulation relation to verify an assertion graph according to a normal satisfiability criteria," as set forth in claim 8 are not found, either expressly or inherently described.

42390.P9429   -21-

2. Claim 18 Is Not Anticipated by Jain.

Appellant respectfully submits that in the cited reference, each and every element

as set forth in claim 18 is not found, either expressly or inherently described.

Claim 18, for example, sets forth:

> 18.   (Original) The method recited in Claim 17 further comprising:
> computing a symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and
> checking the symbolic simulation relation with a symbolic consequence labeling for the assertion graph on the first symbolic lattice domain according to a normal satisfiability criteria.

As stated above with regard to claim 8, the dissertation of Jain is directed to

methods for verifying oblivious trajectory assertions, which correspond to what the

present application refers to as a "strong satisfiability criteria." In verifying the oblivious

trajectory assertions of Jain, effects are checked against corresponding past and present

causes.

Jain does not disclose a method to compute and check a symbolic simulation

relation according to what the present application discloses as a "normal satisfiability

criteria." But the present application discloses, for example, (p. 12, lines 18-22) that:

> Figure 5a shows iterations of antecedent strengthening of graph 202 on model 101.

The present application further discloses (p. 12, lines 13-26, emphasis supplied)

that:

> Figure 5b shows the final simulation relation resulting from iterations of the method of Figure 3a performed on the antecedent strengthened assertion graph 502 and using model 101. Comparing the final simulation relation labels for each edge, with the consequence set for that edge (as shown in assertion graph 202) indicates whether the model 101 strongly satisfies the strengthened assertion graph 502. ... Therefore model 101 strongly satisfies the antecedent strengthened assertion graph 502, but more importantly model 101 satisfies assertion graph 202 according to normal satisfiability as previously defined.

Thus the example illustrated in Figures 5a and 5b show computing a symbolic

simulation relation for the assertion graph 202 and checking the symbolic simulation

42390.P9429   -22-

relation with a symbolic consequence labeling for the assertion graph according to a normal satisfiability criteria.

The methods Jain discloses for verifying oblivious trajectory assertions do not discuss or suggest a way to compute a symbolic simulation relation and to check the symbolic simulation relation with a symbolic consequence labeling according to a normal satisfiability criteria as set forth in claim 18.

Therefore, Appellant respectfully submits that in the cited reference, a method comprising, "computing a symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and checking the symbolic simulation relation with a symbolic consequence labeling for the assertion graph on the first symbolic lattice domain according to a normal satisfiability criteria," is not found, either expressly or inherently described.

42390.P9429   -23-

3. Claim 4 Is Not Anticipated by Jain.

Appellant respectfully submits that in the cited reference, each and every element as set forth in claim 4 is not found, either expressly or inherently described.

The Final Office Action suggests that the limitations of claim 8 are disclosed by Jain in the Abstract (p. iii) and in the Introduction (p.3).

Appellant respectfully disagrees. Claim 4, for example, sets forth:

4.   (Previously Presented) A computer software product including one or more recordable
            media having executable instructions stored thereon which, when executed by a
            processing device, causes the processing device to:
                initialize a symbolic simulation relation for an assertion graph on a first
            symbolic lattice domain, wherein the assertion graph on the first symbolic lattice
            domain is configurable to express a justification property to verify by computing
   the symbolic simulation relation.

As described with regard to claim 8, Jain limits his verification algorithms to what he refers to as oblivious trajectory assertions (§6.3, Def. 9, par. 2). According to Jain's classification of trajectory assertions, the oblivious trajectory assertion is the least expressive and complex of the trajectory assertions (§5.3.2, par. 2).

Appellant respectfully submits that according to the verification algorithms proposed by Jain, the oblivious trajectory assertions are not configurable to express a justification property for verification.

For example, using the state diagram of a modulo-3 counter presented by Jain (§5.3.3, Fig. 5.3, reproduced below), one might attempt to verify the following: if the system enters state B and does not start in state B, then at the time prior to entering state B, the system must have been in state A. Intuitively, one can see that the above justification property is true for the state diagram of Fig. 5.3 reproduced below.

42390.P9429   -24-

Jain Fig. 5.3
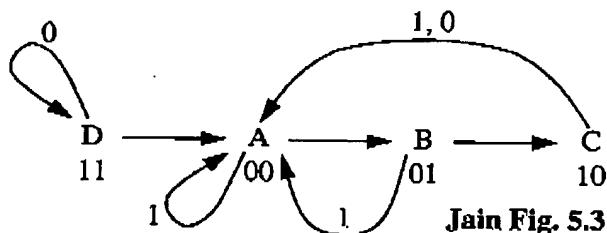
Note that in Jain's example: the values on transitions indicate the value of a *reset* input; the values below the internal states A, B, C and D, represent the values of two internal state variables, which we may refer to as s_1 and s_2; there is one additional input, *in*, and one additional output, *out*; so all possible states may be represented, as Jain does, by tuples of five values, $\langle reset, in, s_1, s_2, out \rangle$.

Trajectory Assertion 1
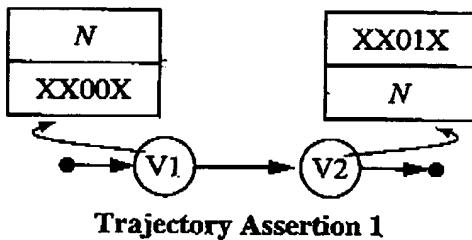
Therefore, defining a simple trajectory assertion according to Jain's proposed methodology, as shown in Trajectory Assertion 1 above, with an action node formula on state vertex V1 to be any of the possible node assignments, $N = \{0,1\}^5 = \{00000, \dots, 11111\}$, and a reaction node formula on V1 to be all possible node assignments for the state A, {00000, 00001, 01000, 01001, 10000, 10001, 11000, 11001}; and further defining an action node formula on the next state vertex V2 to be all possible node assignments for the state B, {00010, 00011, 01010, 01011, 10010, 10011, 11010, 11011}, and a reaction node formula on V2 to be any of the possible node assignments, $N = \{0,1\}^5$; one can check if the verification algorithm shown in Jain's Figure 5.2 would verify that the justification property (which we know, intuitively, is true) holds for the

42390.P9429   -25-

state diagram of Figure 5.3. According to lines 5 and 10 of the algorithm, the defining

trajectory label for the vertex V1 is computed as $\delta(N) \cap N$, which is equal to N. Since

the defining trajectory label for V1 is not contained by the set of reaction assignments for

V1 (e.g. $N \not\subseteq \{00000, 00001, 01000, 01001, 10000, 10001, 11000, 11001\}$), the test

following line 12 in Fig. 5.2 fails, and so verification of the property fails.

The ternary algorithm, of Jain's Fig. 5.8, being more pessimistic than the

algorithm of Fig. 5.2, necessarily, also fails to verify of the property (§5.4.3, last

paragraph). Furthermore, the ternary existential quantification at the end of each iteration

of the algorithm of Jain's Fig. 5.16, line 11, means that the proposed generalized STE

algorithm of Jain is also more pessimistic than the algorithm of Fig. 5.2 and therefore,

also fails to verify of the property (§5.5.2, last paragraph).

Therefore, as the present application discloses (p. 14, line 24 through p. 15, line 1,

emphasis supplied):

> ...what has been demonstrated is that the method proposed by Alok Jain does not provide for justification. In fact, it is substantially impossible to provide for a justification capability within the semantic constraints used by prior STE and GSTE methods.

The satisfiability of oblivious trajectory assertions as proposed by Jain may be

referred to as "strong satisfiability," where effects (reactions) are checked against

corresponding past and present causes (actions). The present application also discloses

(p. 14, lines 4-6) that:

> Strong satisfiability, however, is inadequate for expressing justification properties, which are causes of effects, rather than effects of causes.

The present application further discloses a "normal satisfiablility," which does not

require such strong assumptions with regard to assertion graphs. The present application

discloses (p. 16, lines 4-6, emphasis supplied) that:

42390.P9429   -26-

Therefore, for one embodiment, a <u>normal semantics</u>, herein disclosed, <u>provides for assertion graphs</u>, which are <u>capable of expressing justification properties</u>.

For example, the present application further discloses (p. 21, line 8 through p. 22, line 2, Figs. 6a and 6b) that:

> For one embodiment, Figure 6a shows a method for computing the normal satisfiability of an assertion graph by a model. In block 611, the antecedent sets are strengthened for each edge in the assertion graph. In block 612, a fixpoint simulation relation is computed using the antecedent strengthened assertion graph. Finally in block 613, the simulation relation sets are compared to the consequence sets to see if, for each edge, the simulation relation set is a subset of the consequence set, which is the necessary condition for satisfiability.
>
> For one embodiment, Figure 6b illustrates, in finer detail, a method of computing normal satisfiability. In block 621, the strengthened antecedent set fixpoint for each edge e (denoted Ant*(e)) in assertion graph G is computed. In block 622, a fixpoint simulation relation set for each edge e (denoted Sim*(e)) is computed using the strengthened antecedents computed for each edge in block 621. In block 623, the comparison is performed. First, the edges are marked active in block 624. Then a test is performed in block 625 to determine if any active edges remain to be compared. If not, the method is complete and the assertion graph is satisfied by the model. Otherwise, an active edge, e, is selected in block 626 and set to not active. In block 627, the simulation relation set, Sim*(e), is compared to see if it is a subset of the consequence set, Cons(e). If not, the assertion graph is not satisfied by the model. Otherwise the method flow returns to the test at block 625 to determine if more edges remain to be compared.

Jain does not discuss or suggest an algorithm (e.g. as disclosed with regard to Figs. 6a and 6b of the present application) to initialize a symbolic simulation relation for an assertion graph configurable to express a justification property or to verify a justification property by computing the symbolic simulation relation.

Therefore, Appellant respectfully submits that in the cited reference, instructions which cause a processing device to, "initialize a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify by computing the symbolic simulation relation," as set forth in claim 4 are not found, either expressly or inherently described.

42390.P9429   -27-

4. Claim 14 Is Not Anticipated by Jain.

Appellant respectfully submits that in the cited reference, each and every element

as set forth in claim 14 is not found, either expressly or inherently described.

Claim 14, for example, sets forth:

> 14.   (Previously Presented) A method comprising:
> initializing a symbolic simulation relation for an assertion graph on a first
> symbolic lattice domain, wherein the assertion graph on the first symbolic lattice
> domain is configurable to express a justification property to verify through
> computing the symbolic simulation relation.

As stated above with regard to claim 4, the dissertation of Jain concentrates on

verification algorithms for oblivious trajectory assertions (§6.3, Def. 9, par. 2). These

proposed verification algorithms verify assertions according to what the present

application refers to as a "strong satisfiability criteria."

As demonstrated above with regard to Jain's proposed algorithm of Fig. 5.2 for

verification of trajectory assertion as shown in Fig. 5.4, using the state diagram example

shown in Fig. 5.3, the trajectory assertions of Jain are not configurable to express

justification properties to verify through his proposed algorithm shown in Fig. 5.2.

The present application points out, with regard to the methodologies proposed by

Jain, (p. 12, line 23 through p. 13, line 3) that:

> Strong satisfiability as defined above formally captures a semantics substantially similar to that
> used in STE and GSTE as proposed in 1997 by Alok Jain. It requires that a consequence hold
> based solely on past and present antecedents. Strong satisfiability expresses properties that are
> effects of causes.

The present application also discloses (p. 14, lines 4-6, emphasis supplied) that:

> Strong satisfiability, however, is inadequate for expressing justification properties, which are
> causes of effects, rather than effects of causes.

The present application discloses that a "normal satisfiablility criterea" does not

require such strong assumptions and provides for assertion graphs that are configurable to

42390.P9429   -28-

express justification properties and may be verified through computing the corresponding symbolic simulation relation (e.g. see p. 12, lines 13-26, Fig. 5b; p. 16, lines 4-6 and p. 21, line 8 through p. 22, line 2, Figs. 6a and 6b).

Jain does not discuss or suggest a verification algorithm (e.g. as disclosed with regard to Figs. 6a and 6b of the present application) to initialize a symbolic simulation relation for an assertion graph configurable to express a justification property or to verify a justification property by computing the symbolic simulation relation.

Therefore, Appellant respectfully submits that in the cited reference, a method comprising, "initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify through computing the symbolic simulation relation," as set forth in claim 14 is not found, either expressly or inherently described.

5. Claim 16 Is Not Anticipated by Jain.

Appellant respectfully submits that in the cited reference, each and every element as set forth in claim 16 is not found, either expressly or inherently described.

Claim 16, for example, sets forth:

> 16.   (Original) A method comprising:
>   specifying a justification property with an assertion graph.

Jain makes no mention of a justification property or of how to express such a property with an assertion graph. A justification property is one of the property types that fall into a category of problematic assertions Jain refers to as "prescient." According to Jain, verification of properties in this category would be prohibitively expensive (§6.3, Def. 9, par. 2).

As stated above with regard to claims 4 and 14, the dissertation of Jain concentrates on expressing and verifying oblivious trajectory assertions (§5.3.2, par. 2 and §6.3, Def. 9, par. 2). His proposed verification algorithms can only verify assertions according to what the present application refers to as a "strong satisfiability criteria," which requires a set of strong assumptions.

The present application points out, with regard to the methodologies proposed by Jain, (p. 12, line 23 through p. 13, line 3) that:

> Strong satisfiability as defined above formally captures a semantics substantially similar to that used in STE and GSTE as proposed in 1997 by Alok Jain. It requires that a consequence hold based solely on past and present antecedents. Strong satisfiability expresses properties that are effects of causes.

The present application also discloses (p. 14, lines 4-6, emphasis supplied) that:

> Strong satisfiability, however, is inadequate for expressing justification properties, which are causes of effects, rather than effects of causes.

The present application discloses that a "normal satisfiablility criterea" does not

42390.P9429   -30-

require such strong assumptions and provides for assertion graphs that are configurable to express justification properties and may be verified through computing the corresponding symbolic simulation relation (e.g. see p. 12, lines 13-26, Fig. 5b; p. 16, lines 4-6 and p. 21, line 8 through p. 22, line 2, Figs. 6a and 6b).

Therefore, Appellant respectfully submits that in the cited reference, a method comprising, "specifying a justification property with an assertion graph," as set forth in claim 16 is not found, either expressly or inherently described.

6. Claim 28 Is Not Anticipated by Jain.

Appellant respectfully submits that in the cited reference, each and every element

as set forth in claim 28 are not found, either expressly or inherently described.

Claim 28, for example, sets forth:

28.   (Previously Presented) A verification system comprising:
                means for initializing a symbolic simulation relation for an assertion graph
        on a first symbolic lattice domain, wherein the assertion graph on the first
        symbolic lattice domain is configurable to express a justification property to
        verify through computing the symbolic simulation relation;
                means for computing the symbolic simulation relation for the assertion
        graph on the first symbolic lattice domain; and
                means for checking the symbolic simulation relation to verify a plurality of
        properties expressed by a plurality of corresponding assertion graph instances,
        having at least one assertion graph instance on a second lattice domain different
        from the first symbolic lattice domain.

The dissertation of Jain makes no mention of a justification property or of how to

express such a property with an assertion graph or of how to verify a justification property

through computing the symbolic simulation relation.

The MPEP § 2181 states that:

[U]nless an element performs the identical function specified in the claim, it cannot be an
equivalent for the purposes of 35 U.S.C. 112, sixth paragraph. Pennwalt Corp. v. Durand-
Wayland, Inc., 833 F.2d 931, 4 USPQ2d 1737 (Fed. Cir. 1987), cert. denied, 484 U.S. 961 (1988).

Jain discloses no methods to perform the functions of initializing a symbolic

simulation relation for an assertion graph to express a justification property or of

verifying a justification property through computing the symbolic simulation relation.

For example, the present application discloses (p. 14, lines 6-11; Fig. 2b) that:

As an example of a justification property, one might wish to assert the following: if the system
enters state s1, and does not start in state s1, then at the time prior to entering state s1, the system
must have been in state s0. For one embodiment, Figure 2b depicts an assertion graph 202, which
attempts to capture the justification property asserted in the above example.

The present application further discloses (p. 18, lines 22-25) that:

In order to indicate normal satisfiability, a method is needed to propagate future antecedents

42390.P9429   -32-

backwards. For one embodiment, a method can be defined to strengthen the antecedent set of an edge e by intersecting it with the pre-image sets of antecedents on future edges."

The present application also discloses (p. 20, lines 13-26; Fig. 5b) that:

Figure 5b shows the final simulation relation resulting from iterations of the method of Figure 3a performed on the antecedent strengthened assertion graph 502 and using model 101. Comparing the final simulation relation labels for each edge, with the consequence set for that edge (as shown in assertion graph 202) indicates whether the model 101 strongly satisfies the strengthened assertion graph 502... but more importantly model 101 satisfies assertion graph 202 according to normal satisfiability as previously defined.

Jain does not attempt to verify any properties substantially similar to the justification property set forth in claim 28. As stated above with regard to claims 16, a justification property is one of the property types that fall into a category of problematic assertions Jain refers to as "prescient." According to Jain, verification of properties in this category would be prohibitively expensive (§6.3, Def. 9, par. 2).

As further stated above with regard to claims 4 and 14, the dissertation of Jain concentrates on expressing and verifying oblivious trajectory assertions (§5.3.2, par. 2 and §6.3, Def. 9, par. 2). His proposed verification algorithms can only verify assertions according to what the present application refers to as a "strong satisfiability criteria," which requires a set of strong assumptions.

The present application points out, with regard to the methodologies proposed by Jain, (p. 12, line 23 through p. 13, line 3) that:

Strong satisfiability as defined above formally captures a semantics substantially similar to that used in STE and GSTE as proposed in 1997 by Alok Jain. It requires that a consequence hold based solely on past and present antecedents. Strong satisfiability expresses properties that are effects of causes.

The present application also discloses (p. 14, lines 4-6, emphasis supplied) that:

Strong satisfiability, however, is inadequate for expressing justification properties, which are causes of effects, rather than effects of causes.

The present application discloses that a "normal satisfiablility criterea" does not

42390.P9429   -33-

require such strong assumptions and provides for assertion graphs that are configurable to express justification properties and may be verified through computing the corresponding symbolic simulation relation (e.g. see p. 12, lines 13-26, Fig. 5b; p. 16, lines 4-6 and p. 21, line 8 through p. 22, line 2, Figs. 6a and 6b).

Appellant respectfully submits that in the cited reference, the identical function as set forth in claim 28 is not performed. Therefore, Jain should not be considered equivalent under 35 U.S.C. 112, paragraph six, to the subject matter set forth in claim 28.

Accordingly in light of the argument presented above, Appellant respectfully submits that in the cited reference, a verification system comprising at least, a "means for initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify through computing the symbolic simulation relation," as set forth in claim 28 is not found, either expressly or inherently described.

42390.P9429   -34-

Conclusion

Appellant submits that all claims now pending are in condition for allowance. Such action is earnestly solicited at the earliest possible date. If there is a deficiency in fees, please charge our Deposit Acct. No. 02-2666.

Respectfully submitted,

Date: 7-18-2005
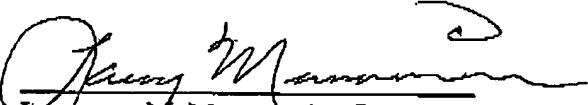
Lawrence M. Mennemeier, Reg. No. 51,003

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8598

42390.P9429   -35-

VIII.   Claims Appendix: Claims Allowed and Involved in Appeal (Clean Copy)

1-3.   (Cancelled)

4.   (Previously Presented) A computer software product including one or more recordable media having executable instructions stored thereon which, when executed by a processing device, causes the processing device to:

    initialize a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify by computing the symbolic simulation relation.

5.   (Original) The computer software product recited in Claim 4 which, when executed by a processing device, further causes the processing device to:

    compute the symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and

    check the symbolic simulation relation to verify a plurality of properties expressed by a plurality of assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

6-7.   (Canceled)

8.   (Previously Presented) A computer software product including one or more recordable media having executable instructions stored thereon which, when executed by a processing device, causes the processing device to:

    initialize a symbolic simulation relation for an assertion graph on a first symbolic lattice domain; and

    compute the symbolic simulation relation for the assertion graph on the first

42390.P9429   -36-

symbolic lattice domain to verify the assertion graph according to a normal satisfiability criteria.

9-13.   (Canceled)

14.   (Previously Presented) A method comprising:

initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify through computing the symbolic simulation relation.

15.   (Original) The method recited in Claim 14 further comprising:

computing the symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and

checking the symbolic simulation relation to verify a plurality of properties expressed by a plurality of corresponding assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

16.   (Original) A method comprising:

specifying a justification property with an assertion graph.

17.   (Original) The method recited in Claim 16 wherein the assertion graph is on a first symbolic lattice domain; and the justification property is expressed by one of a plurality of instances of the assertion graph, at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

18.   (Original) The method recited in Claim 17 further comprising:

computing a symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and

42390.P9429   -37-

checking the symbolic simulation relation with a symbolic consequence labeling for the assertion graph on the first symbolic lattice domain according to a normal satisfiability criteria.

19-27. (Canceled)

28.   (Previously Presented) A verification system comprising:

means for initializing a symbolic simulation relation for an assertion graph on a first symbolic lattice domain, wherein the assertion graph on the first symbolic lattice domain is configurable to express a justification property to verify through computing the symbolic simulation relation;

means for computing the symbolic simulation relation for the assertion graph on the first symbolic lattice domain; and

means for checking the symbolic simulation relation to verify a plurality of properties expressed by a plurality of corresponding assertion graph instances, having at least one assertion graph instance on a second lattice domain different from the first symbolic lattice domain.

29-30. (Canceled)

42390.P9429   -38-